

第15章 网络新闻

网络新闻的起源要追溯到 1979，当时，两个大学生，Tom Truscott和Jim Ellis想到利用UUCP把计算机连接起来，供各用户交换信息之用。于是，他们便于自己就读的北卡罗莱大学内，建立了一个由三台计算机组成的小型网络。

最初，信息的传输是用大量外壳脚本（后来采用C语言重写）来处理的，但这些脚本从来没有公开发布过。因为，它们很快就被“A”新闻替代了，后者是第一个公开发布的新闻软件。

设计“A”新闻软件的时候，并没有打算让它对每天每个新闻组大量文章进行处理。但随着信息量的不断增大，它已经不堪重负，所以Mark Horton和Matt Glickman对它进行了改写，他们把它叫作“B”版（也就是Bnews）。Bnews的第一个版本是1982年发布的2.1。后来人们又对它不断的扩展，并增加了许多新功能。其当前版本是Bnews 2.11。随着其最后一个正式维护人员转向INN，它也逐渐被废弃不用。

对“A”新闻软件的另一次改写完成并发布于1987年，是Geoff Collyer和Henry Spencer改写的，这被称为“C”版。后来还出现了许多C-News补丁，最为突出的是C-News Performance Release。对那些运行许多个新闻组的站点来说，频繁调用中转新闻所涉及的开支是非常之大的，因为要负责分发进入的所有新闻组文章。Performance Release为中转新闻增加了一个选项，允许管理员在后台程序模式下运行它，这种模式下，程序将自己置入后台运行。

Performance Release是当前最新的C-News版本。

C版本的所有新闻主要用于UUCP网络，但也可用于其他环境中。在TCP/IP、DECNet或相关网络上有效地传输新闻，都需要一个新方案。所以NNTP（网络新闻协议）于1986年应运而生。该协议以网络连接为基础，指定了大量的命令，用于交互传输和获取新闻。

网上有许多基于NNTP的应用。其中之一是nntpd包，它是Brian Barber和Phil Lapsley共同编写的，可用于为局域网内的大量主机提供新闻阅读服务。设计nntpd的目的是完善Bnews和C-News之类的新闻包，为它们赋予NNTP特性。

另一个不同的NNTP包是INN，也称作Internet News。它不只是一个前端，而是独立的新闻系统。它包含一个复杂的新闻中转后台程序，这个程序能够有效维护并发的若干个NNTP链接，因此，它也是许多因特网站点首选的新闻服务器。

15.1 何谓Usenet

关于Usenet，最意外的是它并不属于哪个组织，也没有任何核心网络管理人员。事实上，除了技术说明之外，它只是Usenet知识的一部分，所以你不能确切地定义它是……，只能说它不是……。如果你手中有Brendan Kehoe编写的《因特网中的禅机与艺术》（位于www.cs.indiana.edu/docproject/zen/zen-1.0_toc/html）；或看过他的主页（www.zen.org/~brendan），就能充分了解Usenet的特性。

虽然如此，有人仍然将Usenet定义为由各个独立站点组成的群体，这些站点间彼此交换

Usenet新闻。要想成为 Usenet 站点，你只须找到另一个 Usenet 站点，与其拥有者和维护人员达成一致，允许他们与你交换 Usenet 新闻即可。为另一个站点提供新闻，被称为“为它发送、分配或交换新闻”，套用另一句大家熟悉的 Usenet 名言：“提供新闻，你就能上 Usenet 了”。

Usenet 新闻的基本单元是文章。即用户写入或“投递”到网上的消息。为了使新闻系统能够处理它们，所以在这些文章前面加了一些管理信息，也就是所谓的“文章头”。它与因特网邮件标准 RFC-822 中规定的邮件头极为相似，由若干个文本行组成，每行的格式是这样的：字段名:字段值。

注意 Usenet 新闻消息格式是在 RFC-1036 “USENET 消息交换标准”中指定的。

新闻组文章被提交到一个或多个新闻组。我们可以把新闻组想像为一个聚集了与某一主题相关的文章的场所。所有的新闻组都是以分层式结构来组织的，各组的组名便代表它在这个结构中的位置。这样，一眼就能看出一个新闻组的主题是什么。比如，任何人都可从 comp.os.linux.announce 这个新闻组名得知，该新闻组用于发布名为 Linux 的计算机操作系统之相关消息。

提交之后，这些新闻组文章便在愿为该组传播新闻的所有 Usenet 站点间进行交换。两个站点同意交换新闻后，便可根据自己喜好，自由交换新闻组，甚至可加入自己的本地新闻结构。例如，groucho.edu 有一条指向 barnyard.edu（是一个主要的新闻发送站点）和若干条指向次要新闻发送站点的链接。现在，Barnyard 学院可能收到了所有的 Usenet 新闻组，而 GMU 只想传输少数主要的新闻结构比如（sci、comp、rec 等）。有的下游站点，比如说一个叫做 brewhq 的 UUCP 站点，打算传输比前者还要少的新闻组，因为它们缺乏网络和硬件资源。另一方面，brewhq 可能想收到 fj 结构中的新闻组，GMU 却没有将这些新闻组传输过来。因此，它维护了另一条到 gargleblaster.com 的链接，后者将传输所有的 fj 新闻组结构，并把它们发送给 brewhq。

15.2 Usenet 如何对新闻加以控制

如今，Usenet 增长比例日趋加大。以传输整个网络新闻的站点为例，它们每天传输的数据量一般都在 3~5GB 之间。这当然要求对新闻加以更多的控制。因此，为大家讲讲多数系统怎样处理 Usenet 新闻是非常必要的。

注意 如果想了解最新的 Usenix 主题会（会上涉及了一些非常有趣的主题），一定要到这里看看：www.infosys.tuwien.ac.at/staff/pooh/papers/NewsCacheHP/。

新闻在网络间的分发是由不同的传输点来完成的。过去常用的是 UUCP，现在则主要由因特网站点负责传输。采用的路由被称作“扩散式路由选择”（flooding）。每个站点维护许多指向其他站点的链接（新闻发送），本地新闻系统发出或收到的任何一篇文章都将被转发到链接站点，但如果链接站点上已有这些文章的话，它们就会被丢弃。通过查看路径：文章头字段，站点就可以找到文章经历了哪些站点。文章头中包含一个所有系统列表，通过 bang 路径表达式可看出文章经历了哪些系统。

为了区分文章和识别文章是否重复，Usenet 文章必须传输一个消息 ID（是 Message-ID:header 字段中指定的），这个 ID 由发送站点名和一个序列号组合而成，即 < serial@site >。针对每篇处理过的文章，新闻系统都会把这个 ID 记录在历史文件内。要检查所有新到的文章，

可查看这个历史文件。

任何两个站点间的通信流都要受两个标准的限制：其一，一篇文章对应一个分配（定义在Distribution:header字段内，用于把该文章的发送限定在特定的站点组内。其二，参与交换的新闻组要受到发送和接收系统双方的限制。允许传到某一个站点的新闻组集和分配通常保存在sys文件内。

文章的绝对编号通常要求对前面提到的方案进行改进。UUCP网络本身用于定期收集文章并把收集到的文章归入一个单一的文件内。该文件再被压缩，发送到远程站点。这就是所谓的“批处理法”（batching）。

另一种方法是采用ihave/sendme协议，它可防止重复性的文章从本地传输到远程站点，从而节省网络带宽。它采取的作法不是把所有文章放入batch文件，并将它们一并发送到远程站点，而是，只将文章的消息ID合成一条大型的“ihave”消息，再将它发送到远程站点。然后，读取这条消息，把它和自己的历史文件进行比较，并在返回的“sendme”消息中，列出自己需要的文章列表。最后，只有消息中的那些文章才被发送到远程站点。

当然，ihave/sendme只适用于这种情况：交换新闻的两个大型站点各自接收数个发送站点发来的文章，两者频繁交换文章。

因特网上的站点一般依靠基于TCP/IP的软件来进行传输。它采用的是“新闻传输协议”（NNTP，具体说明参见RFC-977）。它在新闻发送双方进行新闻的传输，并提供对远程主机上的个体用户的访问。

NNTP可识别三种新闻传输方式。其一是ihave/sendme的实时版本，也称为“推”（pushing）新闻。其二是“拉”（pulling）新闻，这种方式中，客户机请求一份列出指定新闻组或结构中文章的列表（这些文章是在指定日期后到达服务器站点的），并选出自己历史文件内没有的那些文章。第三种方式则用于交互式阅读，允许你或你的新闻阅读机获取指定新闻组的文章和投递文章头信息不完整的文章。

每个站点上，新闻都是保存在/var/spool/news下面的一个目录结构内，每篇文章在一个单独的文件内，每个新闻组在一个单独的目录内。目录名由新闻组名和路径部分组成。因此，comp.os.linux.misc文章被保存在/var/spool/news/comp/os/linux/misc内。新闻组内的文章根据其到达顺序，被编了号。这个编号被当作文件名使用。目前，在线文章的编号范围被保存在一个名为active的文件内，该文件同时充当你的站点所知的新闻组列表。由于磁盘空间是有限的，所以隔段时间，就应该删除某些文章。这就是所谓的“期满”。通常情况下，来自特定新闻组和结构的文章都会在它们抵达之后的既定天内期满。这个期限可由投递者改写，具体做法是在文章头的Expires:field字段内，指定期满日期。